In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize = (8,5))
```

Out[1]: <Figure size 800x500 with 0 Axes>

<Figure size 800x500 with 0 Axes>

In [2]:
```
df = pd.read_csv("train.csv")
df.head()
```

Out[2]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Co
0	1	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	U
1	2	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	U
2	3	CA-2017-138688	12/06/2017	16/06/2017	Second Class	DV-13045	Darrin Van Huff	Corporate	U
3	4	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	U
4	5	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	U

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9800 entries, 0 to 9799
Data columns (total 18 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Row ID         9800 non-null   int64  
 1   Order ID       9800 non-null   object 
 2   Order Date     9800 non-null   object 
 3   Ship Date      9800 non-null   object 
 4   Ship Mode      9800 non-null   object 
 5   Customer ID    9800 non-null   object 
 6   Customer Name  9800 non-null   object 
 7   Segment        9800 non-null   object 
 8   Country        9800 non-null   object 
 9   City           9800 non-null   object 
 10  State          9800 non-null   object 
 11  Postal Code    9789 non-null   float64
 12  Region         9800 non-null   object 
 13  Product ID     9800 non-null   object 
 14  Category       9800 non-null   object 
 15  Sub-Category   9800 non-null   object 
 16  Product Name   9800 non-null   object 
 17  Sales          9800 non-null   float64
dtypes: float64(2), int64(1), object(15)
memory usage: 1.3+ MB
```

In [4]:
```
df.describe()
```

Out[4]:

	Row ID	Postal Code	Sales
count	9800.000000	9789.000000	9800.000000
mean	4900.500000	55273.322403	230.769059
std	2829.160653	32041.223413	626.651875
min	1.000000	1040.000000	0.444000
25%	2450.750000	23223.000000	17.248000
50%	4900.500000	58103.000000	54.490000
75%	7350.250000	90008.000000	210.605000
max	9800.000000	99301.000000	22638.480000

In [10]:
```
df.columns
```

Out[10]:
```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
       'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',
       'Product Name', 'Sales'],
      dtype='object')
```

In [12]:
```
df["Order Date"] = pd.to_datetime(df["Order Date"], dayfirst=True)
df["Ship Date"] = pd.to_datetime(df["Ship Date"], dayfirst=True)

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9800 entries, 0 to 9799
Data columns (total 18 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9800 non-null   int64
 1   Order ID       9800 non-null   object
 2   Order Date     9800 non-null   datetime64[ns]
 3   Ship Date      9800 non-null   datetime64[ns]
 4   Ship Mode      9800 non-null   object
 5   Customer ID    9800 non-null   object
 6   Customer Name  9800 non-null   object
 7   Segment        9800 non-null   object
 8   Country        9800 non-null   object
 9   City           9800 non-null   object
 10  State          9800 non-null   object
 11  Postal Code    9789 non-null   float64
 12  Region         9800 non-null   object
 13  Product ID     9800 non-null   object
 14  Category       9800 non-null   object
 15  Sub-Category   9800 non-null   object
 16  Product Name   9800 non-null   object
 17  Sales          9800 non-null   float64
dtypes: datetime64[ns](2), float64(2), int64(1), object(13)
memory usage: 1.3+ MB
```

In [14]:
```
total_sales = df["Sales"].sum()
print("Total Sales:", total_sales)
```

```
Total Sales: 2261536.7827000003
```

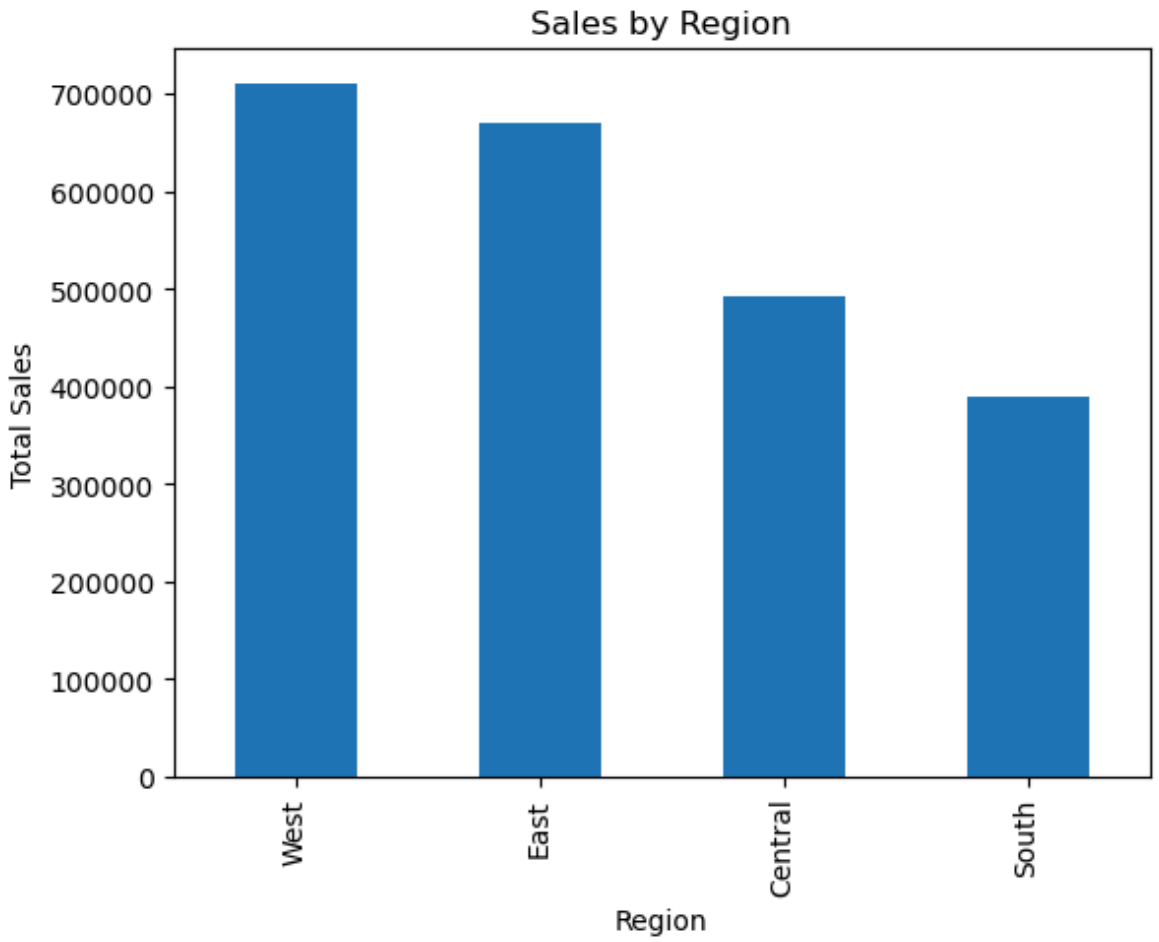
In [15]:
```
total_orders = df["Order ID"].nunique()
print("Total Orders:", total_orders)
```

```
Total Orders: 4922
```

In [16]:
```
region_sales = df.groupby("Region")["Sales"].sum().sort_values(ascending=False)
print(region_sales)
```

```
Region
West       710219.6845
East       669518.7260
Central    492646.9132
South      389151.4590
Name: Sales, dtype: float64
```

In [17]:
```
region_sales.plot(kind="bar")
plt.title("Sales by Region")
plt.xlabel("Region")
plt.ylabel("Total Sales")
plt.show()
```

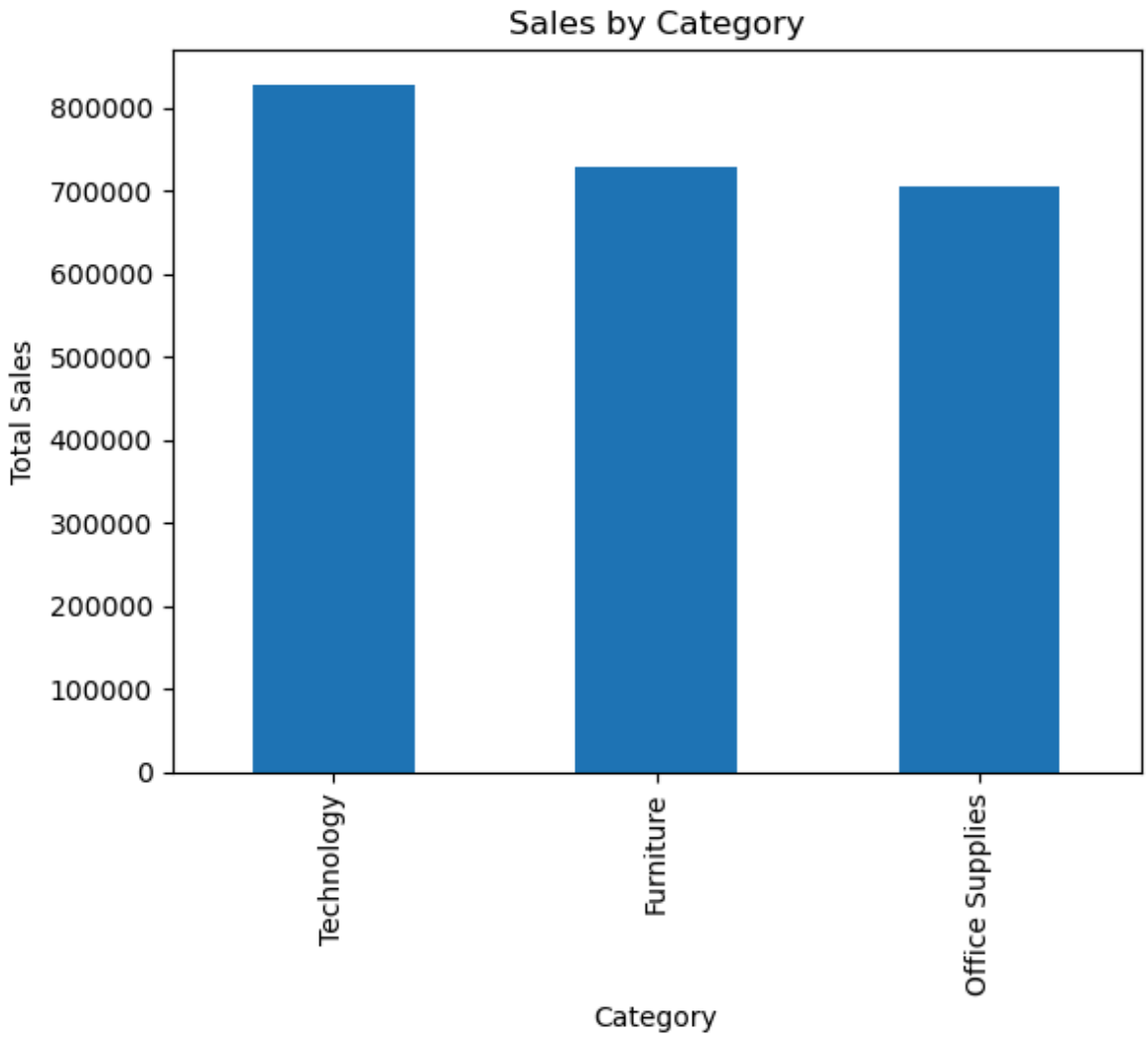


The West region generated the highest sales, indicating strong market penetration and customer demand compared to other regions.

In [18]:
```
category_sales = df.groupby("Category")["Sales"].sum().sort_values(ascending=Fal
print(category_sales)
```

```
Category
Technology         827455.8730
Furniture          728658.5757
Office Supplies    705422.3340
Name: Sales, dtype: float64
```

In [19]:
```
category_sales.plot(kind="bar")
plt.title("Sales by Category")
plt.xlabel("Category")
plt.ylabel("Total Sales")
plt.show()
```

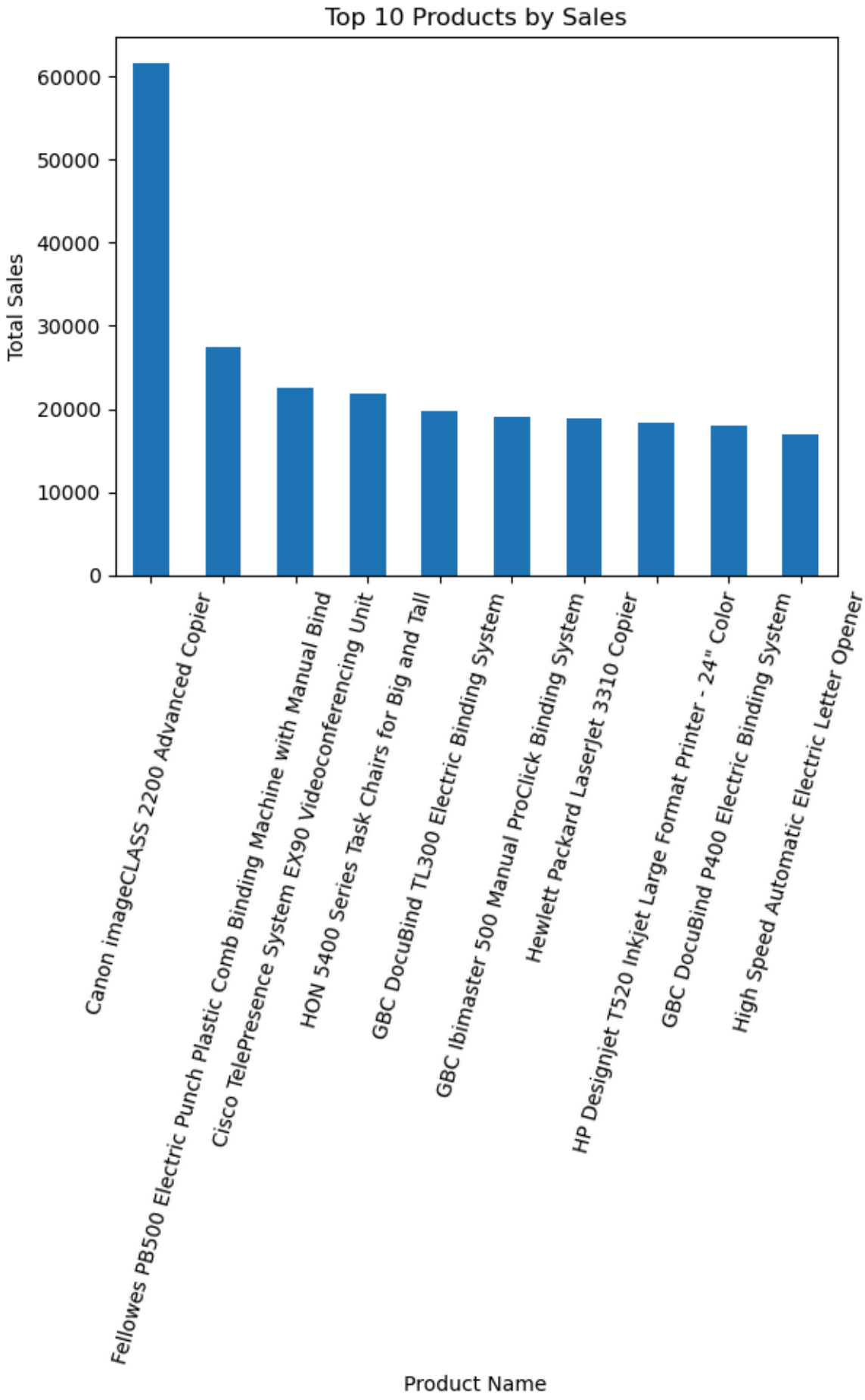


Technology category contributed the highest revenue among all categories, indicating strong customer preference for tech products.

In [20]:
```
top_products = df.groupby("Product Name")["Sales"].sum().sort_values(ascending=F
print(top_products)
```

```
Product Name
Canon imageCLASS 2200 Advanced Copier                                    61
599.824
Fellowes PB500 Electric Punch Plastic Comb Binding Machine with Manual Bind    27
453.384
Cisco TelePresence System EX90 Videoconferencing Unit                    22
638.480
HON 5400 Series Task Chairs for Big and Tall                             21
870.576
GBC DocuBind TL300 Electric Binding System                               19
823.479
GBC Ibimaster 500 Manual ProClick Binding System                         19
024.500
Hewlett Packard LaserJet 3310 Copier                                     18
839.686
HP Designjet T520 Inkjet Large Format Printer - 24" Color                18
374.895
GBC DocuBind P400 Electric Binding System                                17
965.068
High Speed Automatic Electric Letter Opener                              17
030.312
Name: Sales, dtype: float64
```

In [21]:
```
top_products.plot(kind="bar")
plt.title("Top 10 Products by Sales")
plt.xlabel("Product Name")
plt.ylabel("Total Sales")
plt.xticks(rotation=75)
plt.show()
```



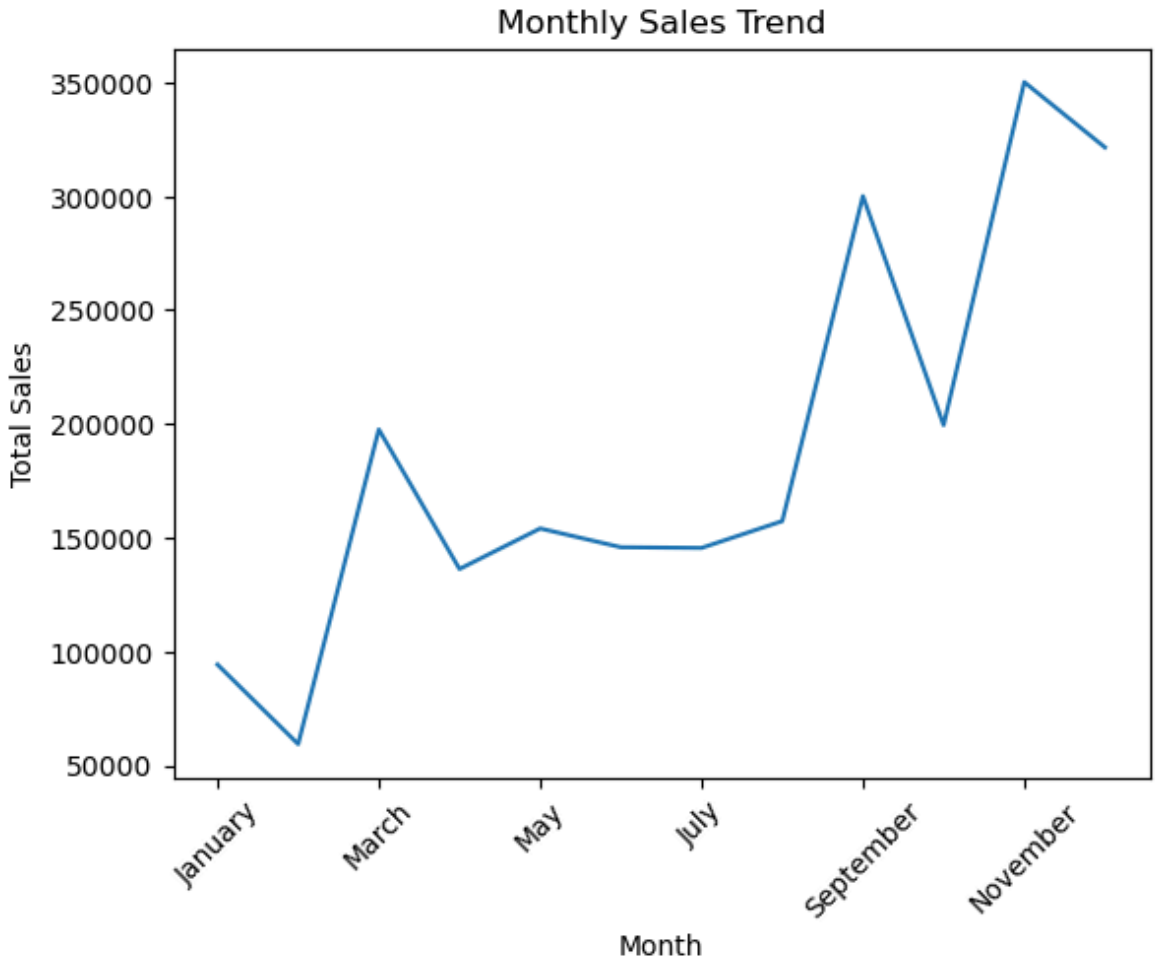
The product "Canon imageCLASS 2200 Advanced Copier" generated the highest sales revenue, indicating strong demand for high-value office equipment.

In [23]:
```
df["Month Name"] = df["Order Date"].dt.month_name()

monthly_sales_name = df.groupby("Month Name")["Sales"].sum()

monthly_sales_name = monthly_sales_name.reindex([
    "January","February","March","April","May","June",
    "July","August","September","October","November","December"
])

monthly_sales_name.plot(kind="line")
plt.title("Monthly Sales Trend")
plt.xlabel("Month")
plt.ylabel("Total Sales")
plt.xticks(rotation=45)
plt.show()
```

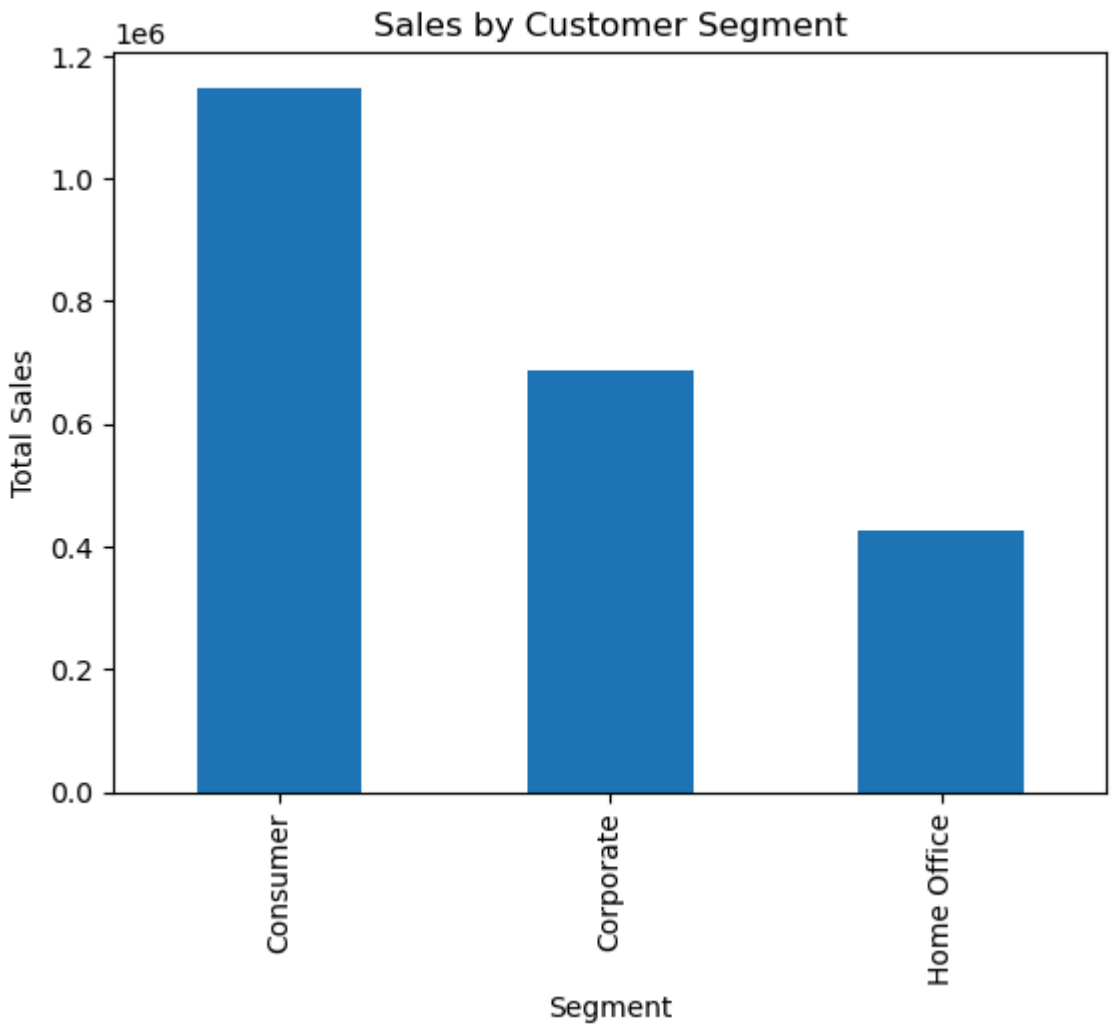


Sales peak observed in november (~350,000), indicating strong year-end demand, possibly driven by holiday season and corporate budget utilization.

In [25]:
```
segment_sales = df.groupby("Segment")["Sales"].sum().sort_values(ascending=False
print(segment_sales)
```

```
Segment
Consumer       1.148061e+06
Corporate      6.884941e+05
Home Office    4.249822e+05
Name: Sales, dtype: float64
```

In [26]:
```
segment_sales.plot(kind="bar")
plt.title("Sales by Customer Segment")
plt.xlabel("Segment")
plt.ylabel("Total Sales")
plt.show()
```



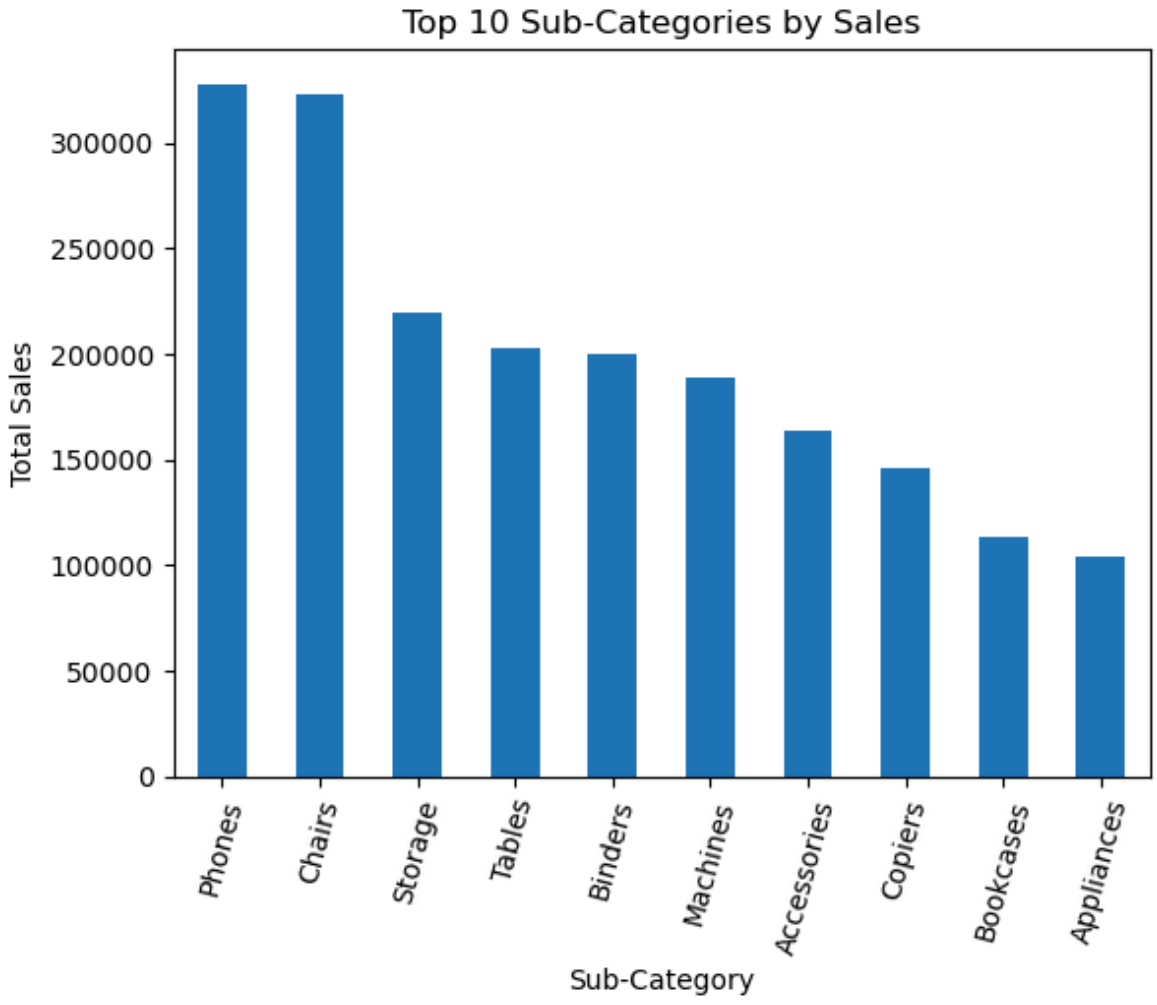
Consumer segment contributes the highest revenue, indicating strong individual customer demand compared to corporate and home office segments.

In [27]:
```
sub_category_sales = df.groupby("Sub-Category")["Sales"].sum().sort_values(ascen

print(sub_category_sales)
```

```
Sub-Category
Phones          327782.4480
Chairs          322822.7310
Storage         219343.3920
Tables          202810.6280
Binders         200028.7850
Machines        189238.6310
Accessories     164186.7000
Copiers         146248.0940
Bookcases       113813.1987
Appliances      104618.4030
Furnishings      89212.0180
Paper            76828.3040
Supplies         46420.3080
Art              26705.4100
Envelopes        16128.0460
Labels           12347.7260
Fasteners         3001.9600
Name: Sales, dtype: float64
```

In [28]:
```
sub_category_sales.head(10).plot(kind="bar")
plt.title("Top 10 Sub-Categories by Sales")
plt.xlabel("Sub-Category")
plt.ylabel("Total Sales")
plt.xticks(rotation=75)
plt.show()
```



Overall Performance

The dataset contains 9,800 transactions with significant overall revenue generation, indicating a stable retail sales performance.

Regional Performance

The West region generated the highest total sales, suggesting strong market penetration and customer demand in that region.

Category Analysis

Technology emerged as the top-performing category, contributing the highest share of revenue among all product categories.

Top Product

The product "Canon imageCLASS 2200 Advanced Copier" recorded the highest sales revenue, indicating strong demand for high-value office equipment.

Monthly Trend

Sales peaked in December (~350,000), indicating strong year-end demand, possibly driven by holiday season and corporate budget utilization.

Customer Segment

The Consumer segment contributed the highest revenue, indicating strong individual customer demand compared to other segments.

Sub-Category

Among sub-categories, Phones generated the highest revenue, suggesting strong performance in that product segment.

Conclusion

Overall, the analysis highlights that revenue is primarily driven by the West region, Technology category, and strong year-end sales trends, providing valuable insights for strategic business decision-making